

# LDA-based Document Model for Adhoc-Retrieval

Andreas Rudolf

Martin Luther Universität  
Halle-Wittenberg

30. März 2007

# Inhaltsverzeichnis

- 1 Einführung
- 2 klassische Verfahren
  - pLSI
  - Clusterbasiertes Retrieval
- 3 LDA-basierte Modelle
  - Latent Dirichlet Allocation
  - LDA-basiertes Retrieval
  - Komplexität
- 4 Experimente und Ergebnisse
  - Feineinstellung
  - Parameter
  - Ergebnisse

# bisherige Verfahren

- Dokumente werden als „**bag of words**“ dargestellt
  - Wörter werden als unabhängig betrachtet
- Topic-Modelle
  - Word- / Document-clustering verbessern die Darstellung
  - **Latent Semantic Indexing** (Deerwester et al 1990)
    - SVD
    - Dimensionsreduktion zahlt sich bei grossen Sammlungen aus
    - Rauschunterdrückung
  - **probabilistic LSI** (Hoffmann 1999)
    - geht von verborgenen Variablen aus
    - Dokumente entstehen durch verschiedene Topics

# Latent Dirichlet Allocation

- inspirierte die Forschung im Bereich Maschinelles Lernen
- als effektiv erwiesen in text-bezogenen Aufgaben (Klassifikation)
- Durchführbarkeit? Effektivität?
- vollständig generative Semantik
- beseitigt Problem des pLSI

# pLSI

- auch „Aspect Model“
- Weiterentwicklung von LSI (textbezogener)
- verbindet jedes gelesene Wort mit einer verborgenen Variable
- Schwächen der Annahme im Unigrammischmodell:  
jedes Dokument wird von genau einer Topic erzeugt

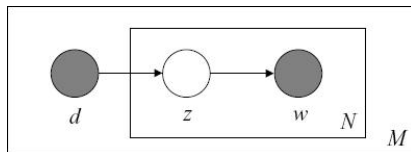
# Prozess

- 1 Wähle eine Topic-verteilung  $P(\cdot | d)$  für jedes Dokument  $d$
- 2 Wähle eine Topic  $z$  mit  $P(z | d)$  für jedes Wort  $w$
- 3 Erzeuge das Wort  $w$  mit  $P(w | z)$

Wahrscheinlichkeit ein Dokument  $d = (w_1, \dots, w_{N_d})$  zu erzeugen:

$$P(w_1, \dots, w_{N_d}) = \prod_{i=1}^{N_d} \sum_{z=1}^K P(w_i | z) \cdot P(z | d)$$

# Prozess als Grafik



## Nachteile

- bessere Performanz als LSI, aber
  - Testsammlungen mit 1033, 1400, 3204, 1460 Dokumenten
  - von Hand optimierte Parameter
- unklare generative Semantik
- Anzahl der Parameter ist proportional zur Grösse der Trainingsammlung  $\implies$  Overfitting

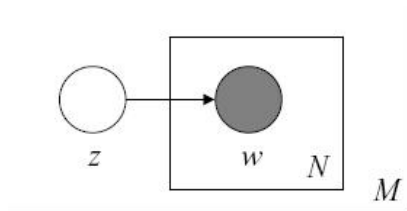
# Unigrammischmodell

- jedes Dokument liegt in einem der  $k$  Cluster
- jeder Cluster beschäftigt sich mit einer Topic  $z$
- jedes  $z$  ist verbunden mit einer Multinomialverteilung  $P(w | z)$  über dem Vokabular
- Prozess
  - 1 Wähle ein  $z$  aus einer Multinomialverteilung mit dem Parameter  $\theta_z$
  - 2 For  $i = 1, \dots, N_d$  wähle das Wort  $w_i$  aus  $z$  mit  $P(w_i | z)$

Wahrscheinlichkeit ein Dokument  $d = (w_1, \dots, w_{N_d})$  zu erzeugen:

$$P(w_1, \dots, w_{N_d}) = \sum_{z=1}^K P(z) \cdot \prod_{i=1}^{N_d} P(w_i | z)$$

# Prozess als Grafik



# Unigrammischmodell - Parameter

- 1 Fasse die Dokumente in  $k$  Gruppen / Cluster zusammen (z.B. K-Means-Algorithmus)
- 2 bestimme mittels Maximum-Likelihood je ein Topic-Modell  $P(w | z)$

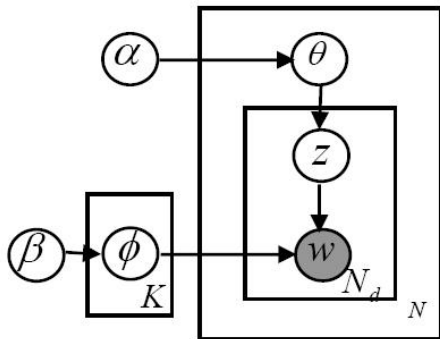
## Wahrscheinlichkeit mit Glättung

$$P(w | D) = \frac{N_d}{N_d + \mu} \cdot P_{ML}(w | D) + \left(1 - \frac{N_d}{N_d + \mu}\right) \cdot P(w | cluster)$$

# LDA-Verfahren

- Generative Prozess ähnelt stark pLSI
- Unterschied :
  - pLSI macht die Topicverteilung abhängig von jedem Dokument
  - LDA bezieht die Topicverteilung aus einer Dirichletverteilung, die für alle Dokumente gleich ist
- Prozess :
  - 1 Wähle eine Multinomialverteilung  $\phi_z$  für jede Topic  $z$  aus einer Dirichletverteilung mit dem Parameter  $\beta$
  - 2 Für jedes Dokument  $d$ , wähle eine Multinomialverteilung  $\theta_d$  aus einer Dirichletverteilung  $\alpha$
  - 3 Für jedes Wort  $w$  im Dokument  $d$ , wähle eine Topic  $z \in 1, \dots, K$  aus der Multinomialverteilung  $\theta_d$
  - 4 Wähle ein Wort  $w$  aus der Multinomialverteilung

# Prozess als Grafik



# LDA-Verfahren

Wahrscheinlichkeit eine Textsammlung zu generieren

$$P(\text{doc}_1, \dots, \text{doc}_N \mid \alpha, \beta) =$$

$$\int \int \prod_{z=1}^K P(\phi_z \mid \beta) \cdot \prod_{d=1}^N P(\theta_d \mid \alpha) \cdot \left( \prod_{i=1}^{N_d} \sum_{z_i=1}^K P(z_i \mid \theta) \cdot P(w_i \mid z, \phi) \right) d\theta d\phi$$

# Vergleiche zu

- pLSI
  - vollständige generative Semantik
  - kein Overfittingproblem
- cluster-basiertes Modelle
  - erlaubt, dass ein Dokument durch verschiedene Topics erzeugt wurde
  - mehr dazu später

Anfrage  $Q = (q_1, \dots, q_n)$

$$P(Q | D) = \prod_{q \in Q} P(q | D)$$

$P(q_i | D)$  gegeben durch ( $\mu = 1000$ )

$$P(w | D) = \frac{N_d}{N_d + \mu} \cdot P_{ML}(w | D) + \left(1 - \frac{N_d}{N_d + \mu}\right) \cdot P_{ML}(w | coll)$$

- LDA : Topics werden als Kombination von Wörtern dargestellt  
⇒ könnte nicht so präzise sein wie in Nicht-Topic-Modellen  
(Unigrammodell)

### Kombination des Originalverfahrens mit dem LDA

$$P(w | D) =$$

$$\lambda \left( \frac{N_d}{N_d + \mu} \cdot P_{ML}(w | D) + \left( 1 - \frac{N_d}{N_d + \mu} \right) \cdot P_{ML}(w | coll) \right) +$$
$$(1 - \lambda) \cdot P_{lda}(w | D)$$

# Kombination des Originalverfahrens mit dem LDA

- a posteriori Wahrscheinlichkeiten von  $\theta$  und  $\phi$  sind zu bestimmen ( $\hat{\theta}$  und  $\hat{\phi}$ )

## Wahrscheinlichkeit eines Wortes $w$

$$P_{lda}(w | d, \hat{\theta}, \hat{\phi}) = \sum_{z=1}^K P(w | z, \hat{\phi}) \cdot P(z | \hat{\theta}, d)$$

- kann nicht exakt gelöst werden
- Näherungsverfahren
  - Expectation Propagation
  - Variations-Methoden
  - hier Gibbs-Sampling

# Gibbs-Sampling

- $n_{-i,j}^{w_i}$  : Anzahl der Instanzen von  $w_i$ , die der Topic  $z=j$  zugewiesen wurde (ohne die Instanz  $w_i$ )
- $n_{-i,j}^{d_i}$  : Anzahl der Wörter in  $d_i$
- $\sum_{v=1}^V n_{-i,j}^v$  : Gesamtzahl der Wörter in Topic  $z=j$
- $\sum_{t=1}^T n_{-i,t}^{d_i}$  : Gesamtzahl der Wörter in  $d_i$
- $\alpha, \beta$  Glättungsparameter

# Gibbs-Sampling

nach einigen Iterationen gilt als Näherung

$$\hat{\phi} = \frac{(n_{-i,j}^{w_i} + \beta_{w_i})}{\sum_{v=1}^V (n_{-i,j}^v + \beta_v)}$$

$$\hat{\theta} = \frac{(n_{-i,j}^{d_i} + \alpha_{z_i})}{\sum_{t=1}^T (n_{-i,t}^{d_i} + \alpha_t)}$$

## Kombination des Originalverfahrens mit dem LDA

$$P(w | D) =$$

$$\lambda \left( \frac{N_d}{N_d + \mu} \cdot P_{ML}(w | D) + \left( 1 - \frac{N_d}{N_d + \mu} \right) \cdot P_{ML}(w | coll) \right) +$$

$$(1 - \lambda) \cdot \left( \sum_{z=1}^K \frac{\left( n_{-i,j}^{w_i} + \beta_{w_i} \right)}{\sum_{v=1}^V \left( n_{-i,j}^v + \beta_v \right)} \times \frac{\left( n_{-i,j}^{d_i} + \alpha_{z_i} \right)}{\sum_{t=1}^T \left( n_{-i,t}^{d_i} + \alpha_t \right)} \right)$$

- Anmerkung : Wert für  $P_{lda}$  stellt eine Mittelung über mehrere Markov-Ketten dar

# Komplexität

- Gibbs-sampling ist linear in  $I, K, N \cdot \bar{N}_t$ 
  - $I$  : Iteration
  - $K$  : Anzahl der Topics
  - $N$  : Anzahl der Dokumente
  - $\bar{N}_t$  : Durchschnittliche Wörteranzahl pro Dokument
- K-Means ist linear in  $I, N, K \cdot \bar{N}_w$ 
  - $\bar{N}_w$  : Durchschnittliche Anzahl einmaliger Terme pro Cluster

# Komplexität

- $K$  : Anzahl Topics in LDA < Anzahl Cluster in Cluster-Modell
- $I$  : Anzahl Iterationen in LDA > 3-pass k-Means-Algorithmus (30 - 50 für Markovketten notwendig)
- $\bar{N}_t$  vs  $\bar{N}_w$  :
  - $\bar{N}_w$  hängt vom gewählten  $K$  ab
  - in Experimenten ist oft  $\bar{N}_w > \bar{N}_t$ , da ein Cluster viele Dokumente enthält
- Experimente ergaben keine sichtlichen Unterschiede in der Laufzeit, beide  $O(K \cdot N)$

# Datenbasis

- gleiche Sammlung wie bei Liu und Croft
  - ausser „Federal Register“ : zu wenig relevante Anfragen
- Anfragen stammen aus dem Titel der TREC-Topics
- Anfragen ohne relevante Dokumente wurden entfernt

# Datenbasis

Collection	Contents	# of dos	Size	Queries	# of Queries with Relevant Docs
AP	Associated Press newswire 1988-90	242,918	0.73Gb	TREC topics 51-150 (title only)	99
FT	Financial Times 1991-94	210,158	0.56Gb	TREC topics 301-400 (title only)	95
SJMN	San Jose Mercury News 1991	90,257	0.29Gb	TREC topics 51-150 (title only)	94
LA	LA Times	131,896	0.48Gb	TREC topics 301-400 (title only)	98
WSJ	Wall Street Journal 1987-92	173,252	0.51Gb	TREC topics 51-100 & 151-200 (title only)	100

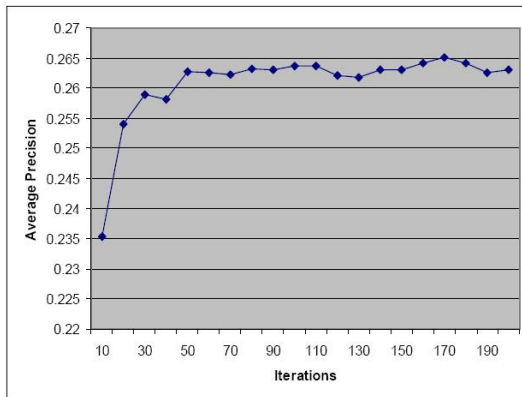
# Parameterwahl

- Uns interessieren
  - $\lambda$  : Anteil des LDA-Teil
  - $l$  : Anzahl der Iterationen
  - Anzahl der Markov-Ketten
- Parameter wurden von Hand auf beste Durchschnittliche Precision optimiert (Retrieval)
- gleiche Trainingsammlung : „Associated Press“
- $\alpha = 50/K$ ,  $\beta = 0.01$  (Standardwerte) : kaum Einfluss auf Ergebnisse?

# Markov-Kette

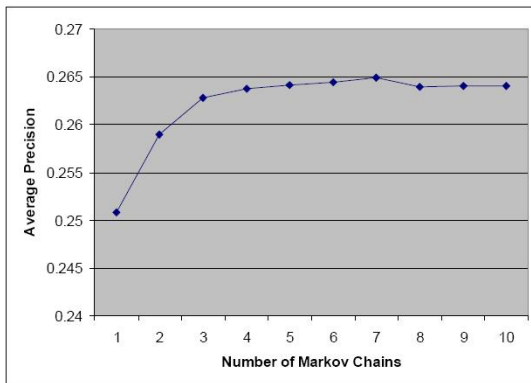
- Problem : Wann ist Markov-Kette stabil?
  - keine Vorhersage möglich
  - viele Iterationen werden als ausreichend angesehen
  - nicht praktikabel bei grossen Datenmengen

# Markov-Kette



**Figure 2. Retrieval results (in average precision) on AP with different number of iterations.  $K=400$ ;  $\lambda=0.7$ ; 1 Markov chain.**

# Markov-Kette



**Figure 3. Retrieval results (in average precision) on AP with different number of Markov chains.  $K=400$ ;  $\lambda =0.7$ ; 30 iterations.**

# K wählen

- K bestimmen mit z.B. Chinese-Restaurant-Prozess nicht möglich
- Allgemein genügt  $50 \leq K \leq 300$
- Optimal :
  - Clusterverfahren :  $K = 2000$
  - LDA :  $K = 800$  (siehe Abb.)

## K wählen

<b><math>K</math></b>	<b>100</b>	<b>200</b>	<b>300</b>	<b>400</b>	<b>500</b>
Average precision	0.2431	0.2520	0.2579	0.2590	0.2557
<b>600</b>	<b>700</b>	<b>800</b>	<b>900</b>	<b>1000</b>	<b>1500</b>
0.2578	0.2609	0.2621	0.2613	0.2585	0.2579

# Retrievalparameter

beste Ergebnisse mit

- $\mu = 1000$
- $\lambda = 0.7$

	QL	CBDM	LBDM	%chg over QL	%chg over CBDM
Rel.	21819	21819	21819		
Rel. Retr.	10130	10751	12064	+19.09*	+12.21*
0.00	0.6422	0.6485	0.6795	+5.8*	+4.8*
0.10	0.4339	0.4517	0.4844	+11.6*	+7.2*
0.20	0.3477	0.3713	0.4131	+18.8*	+11.2*
0.30	0.2977	0.317	0.3661	+23.0*	+15.5*
0.40	0.2454	0.2668	0.311	+26.8*	+16.6*
0.50	0.2081	0.2274	0.2666	+28.1*	+17.2*
0.60	0.1696	0.1794	0.2245	+32.4*	+25.1*
0.70	0.1298	0.1444	0.1665	+28.3*	+15.3*
0.80	0.0865	0.1002	0.118	+36.5*	+17.8*
0.90	0.0480	0.0571	0.0694	+44.7*	+21.6
1.00	0.0220	0.0201	0.0187	-15.1	-6.8
Avg	0.2179	0.2326	0.2651	+21.64*	+13.97*

Collection	QL	CBDM	LBDM	%chg over QL	%chg over CBDM
AP	0.2179	0.2326	0.2651	+21.64*	+13.97*
FT	0.2589	0.2713	0.2807	+7.54*	+3.46*
SJMN	0.2032	0.2171	0.2307	+13.57*	+6.26*
LA	0.2468	0.2590	0.2666	+8.02 <sup>2</sup>	+2.93
WSJ	0.2958	0.2984	0.3253	+9.97*	+9.01*

( $l = 50$ , 3 Markovketten)

# Vergleich mit Relevance-Model

<b>Collection</b>	<b>QL</b>	<b>LBDM</b>	<b>RM</b>	<b>%diff</b>
<b>AP</b>	0.2179	0.2651	0.2745	-3.42
<b>FT</b>	0.2589	0.2807	0.2835	-0.99
<b>SJMN</b>	0.2032	0.2307	0.2633	-12.38
<b>LA</b>	0.2468	0.2666	0.2614	+0.20
<b>WSJ</b>	0.2958	0.3253	0.3422	-4.94



# LBDM als Pseudo-Feedback

<b>Collection</b>	<b>QL<sup>3</sup></b>	<b>RM</b>	<b>RM+LBDM</b>	<b>%chg over RM</b>
<b>AP</b>	0.2161	0.2758	0.2869	+4.00
<b>FT</b>	0.2558	0.2889	0.2907	+0.62
<b>SJMN</b>	0.1985	0.2547	0.2603	+2.22
<b>LA</b>	0.2290	0.2509	0.2715	+8.21
<b>WSJ</b>	0.2908	0.3405	0.3606	+5.91*